

Uwe Krause

# (wip) Noch kein Titel

**Fakultät Engineering and Computer Science**  
Department Computer Science

# Inhaltsverzeichnis

<b>1 Motivation &amp; Vorhaben</b>	<b>1</b>
<b>2 Didaktische Basis</b>	<b>2</b>
2.1 Blooms Taxonomie . . . . .	2
2.2 ICAP (Interactive - Constructive - Active - Passive) . . . . .	3
2.3 Methoden der Wissensvermittlung . . . . .	4
2.4 Serious Games ermöglichen Interaktion . . . . .	5
<b>3 Interaktive Lernumgebung</b>	<b>5</b>
3.1 Lernziele . . . . .	6
3.1.1 Broken Access Control . . . . .	6
3.2 Szenario . . . . .	7
3.2.1 Glesn . . . . .	7
3.2.2 Aufbau der Anwendung . . . . .	7
3.3 Sicherheitslücken in der Anwendung . . . . .	9
3.3.1 Auslesen von Meta-Daten trotz fehlender Zugriffsberechtigung . . . . .	9
3.3.2 Rechteauserweiterung . . . . .	9
3.4 Angedachter Lösungsweg . . . . .	10
3.5 Beta-Test . . . . .	12
3.6 Durchführung Workshop . . . . .	12
<b>4 Fazit &amp; Ausblick</b>	<b>13</b>
4.1 Fazit . . . . .	13
4.2 Ausblick . . . . .	13
<b>Bibliographie</b>	<b>15</b>

# 1 Motivation & Vorhaben

Erfolgreiche Angriffe auf IT-Systeme verursachen durch Geschäftsunterbrechung und Wiederherstellungszeiten, Reputationsverlust und eventuelle Bußgelder enorme Kosten. Aktuell wird von einer durchschnittlichen Schadenshöhe von 8,79 Millionen<sup>1</sup> US Dollar für einen auf eine Sicherheitslücke zurückführbaren Datenschutzverstoß ausgegangen. [1] Da die Ursachen für Sicherheitslücken in IT-Systemen unsichere Programmierung oder Programmierfehler sind, kann die Verbesserung der Qualität von Programmcode die Eintrittswahrscheinlichkeit eines Datenschutzverstoßes reduzieren.

Eine Möglichkeit der Qualitätsverbesserung – direkt bei den Menschen ansetzend, die täglich potentiell problematischen Code generieren – ist das Schaffen oder Erhöhen eines Problembewusstseins, welche direkten Auswirkungen es haben kann, wenn durch unvorsichtige Programmierung unsicherer Code erzeugt wird. Erleben Programmierer:innen selbst, wie leicht Sicherheitslücken im Programmcode ausnutzbar sind, ist zu erhoffen, dass sie mehr Verständnis für sichere Programmiermethoden aufbauen. Basierend auf der Annahme, dass Lernende am erfolgreichsten sind, wenn sie sich aktiv mit dem Lerngegenstand beschäftigen, ist nach einer geeigneten Methode zu suchen, mit der Programmierer:innen sich möglichst intensiv mit Sicherheitslücken beschäftigen.

Durch Anwendung der ICAP-Methode (Interactive, Constructive, Active, Passive) [2] soll erreicht werden, dass Entwickler:innen ein gemäß Blooms (überarbeiteter) Taxonomy [3] vertieftes Verständnis nicht nur einer einzelnen Sicherheitslücke, sondern einer ganzen Kategorie an Sicherheitsproblemen erhalten. Dies wiederum soll dafür sorgen, dass sie künftig diese und ähnliche Unachtsamkeiten nicht nur nicht mehr in eigenem Code produzieren, sondern auch in fremden Code identifizieren können.

Das Ziel soll zwar nicht die Weiterbildung von Entwickler:innen zu Penetrationstester:innen sein, jedoch lassen sich Elemente aus verbreiteten Trainingsmethoden von IT-Sicherheits-Expert:innen für den Lernprozess der Qualitätsverbesserung entleihen. Als eines der in der Aus- und Weiterbildung von Penetrationstester:innen verwendeten Vorgehen können Übungen nach dem Vorbild von *capture the flag* [4] verwendet werden. Die Anwendung dieser *serious games* [5] zeichnet sich durch einen hohen Interaktionsgrad aus, weshalb davon auszugehen ist, dass investierte Zeit effizient genutzt wird, um das IT-Sicherheits-Problembewusstsein von Entwickler:innen zu schärfen.

---

<sup>1</sup>4,46 Mio. USD „Unbekannte Zero-Day-Sicherheitslücke“ plus 4,33 Mio. USD „Bekanntermaßen ungepatchte Sicherheitslücke“ [1]

Die weitverbreitetste Sicherheitslücke im Internet ist die mangelhafte Zugriffskontrolle [6]. Damit die Auswirkungen von Programmierfehlern dieser Art besser greifbar werden, wird in dieser Arbeit eine absichtlich verwundbare Testumgebung entwickelt. Diese *Challenge* wird in einem Workshop verwendet, um Teilnehmer:innen dieses Problemfeld näherzubringen.

## 2 Didaktische Basis

Die gewünschte Erhöhung des Problembewusstseins entspricht letztlich einem Dazulernen der Programmierer:innen. Verschiedene didaktische Konzepte bekräftigen, dass Lernende am erfolgreichsten sind, wenn sie sich aktiv mit dem Lerngegenstand beschäftigen.

### 2.1 Blooms Taxonomie

Gemäß Blooms (überarbeiteter) Taxonomie werden verschiedene aufeinander aufbauende Ebenen des Verständnisses identifiziert. Diese werden oft wie in Abbildung 1 als Pyramide dargestellt. Die Basis bildet das „Erinnern“, die Fähigkeit relevantes Wissen aus dem Langzeitgedächtnis abzurufen. Darauf aufbauend folgt das „Verstehen“, das es ermöglicht Wissen zusammenzufassen, zu interpretieren, zu vergleichen und zu erklären. Die Stufe des „Anwendens“ enthält das selbstständige Ausführen von Gelerntem. Darauf folgt die Fähigkeit der „Analyse“ von Sachverhalten, also die Zerlegung des Materials in seine einzelnen Komponenten, Identifikation übergreifender Struktur und Zweck sowie das Erkennen von Mustern. Erst wer dazu in der Lage ist, wird überhaupt die Grundlagen haben, um fachlich „bewerten“ zu können, sich ein Urteil zu bilden, bis hin dazu, Bestehendes zu kritisieren. An der Spitze der Taxonomie steht die Fähigkeit angeeignete Elemente zusammenzufügen, um Neues zu erschaffen. [3]

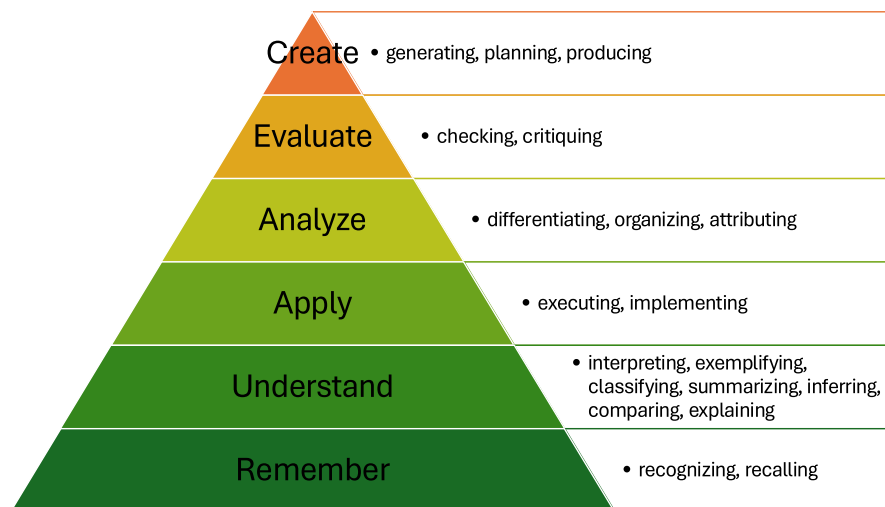


Abbildung 1: Blooms Pyramide, nach [3]. Eigene Darstellung

## 2.2 ICAP (Interactive - Constructive - Active - Passive)

Bei dem *ICAP*-Ansatz werden aufbauend auf dem Vorgehen des *aktiven Lernens* vier Modelle kognitiver Lernaktivitäten definiert und Wege vorgeschlagen, wie Wissensveränderungsprozesse angestoßen werden sollten: Zwar kann Wissen durch passives Empfangen von Informationen beim Zuhören oder Lesen aufgebaut werden, dies stellt jedoch die niedrigste Stufe dar. Die Wissensaufnahme kann aktiver gestaltet werden, zum Beispiel durch das Anfertigen zusätzlicher Notizen während des Zuhörens, dem Anstreichen von Textstellen während des Lesens oder durch Zusammenfassen einzelner Passagen. Konstruktives Verarbeiten erfolgt, wenn über die Aufnahme von Erlerntem hinausgehend selbst etwas generiert wird, beispielsweise wenn in einer Vorlesung Fragen gestellt werden oder zu einem beschriebenen Zusammenhang eine Zeichnung angefertigt wird. Die höchste Stufe des Modells wird erst durch Interaktion mit dem Lerngegenstand erreicht, beispielsweise durch Verteidigung eines Standpunkts in einer Gruppendiskussion, in deren Verlauf neue Ideen entstehen. [2]

Diesen Lernaktivitäten werden die jeweilig zu erwartenden kognitiven Ergebnisse gegenübergestellt: Passiv erlangtes Wissen kann wörtlich und im gleichen Kontext wiedergegeben werden, wohingegen aktiv Erlangtes auf ähnliche Kontexte angewendet werden kann. Einmal generiertes Wissen kann in neuen Situationen angebracht werden, wobei Kenntnis über das erlernte Konzept Interpretation und Erklärungen erlaubt, auch unter

geänderten Rahmenbedingungen. Das tiefste Verständnis ist nach der Interaktion zu erwarten: Potentiell entstehen neue Produkte, Interpretationen und Ideen. [2]

### 2.3 Methoden der Wissensvermittlung

Die Essenz und ein in diversen weiteren Ansätzen<sup>2</sup> immer wieder zu beobachtende Muster ist die möglichst aktive Beschäftigung mit dem Lerngegenstand durch die Lernenden selbst. Wenn Jura-Studierende Gerichtsurteile aufarbeiten, im Betriebswirtschaftsstudium Fallstudien bearbeitet werden und angehende Mediziner:innen anhand der Begutachtung von Leichen die Todesursache ermitteln [7], wie können Programmierer:innen sich interaktiv mit den abstrakten Themen der IT-Sicherheit und sicheren Programmierung beschäftigen?

Wird versucht Konzepte der IT-Sicherheit über Vorträge oder Webinare zu IT-Sicherheit zu vermitteln, stellen diese (sofern sie denn überhaupt wahrgenommen werden; ungeteilte Aufmerksamkeit vorausgesetzt) lediglich die niedrigste Stufe der ICAP-Skala dar. Mehr als eine grundlegende Wissensaufnahme ist nicht zu erwarten, was im Bloomschen Modell ebenfalls der untersten Stufe entspricht. Eine aktivere Beschäftigung mit sicheren Programmiermethoden kann etwa bei *code reviews* das Hervorheben von fehlerhaftem Code sein. Werden Problembereiche auf diese Art erkannt, entspricht dies dem „Verstehen“ auf der Bloomschen Skala. Es darf erwartet werden, dass das Wissen ausreichend abstrahiert wurde, sodass die Chance besteht, ähnliche Fehler in ähnlichen Kontexten erneut zu identifizieren. Um sich konstruktiv mit IT-Sicherheit zu beschäftigen, bieten sich *Hackathons* an, bei denen sicherheitsrelevante Systeme oder Komponenten entwickelt werden. Eine Person, die bereits einmal etwa ein Login-System entworfen oder implementiert hat, wird benötigte Verfahren nicht nur angewendet haben, sondern wahrscheinlich das Wissen auch in seine Bestandteile zerlegt und zumindest Teile davon analysiert haben. Sie sollte diese Fähigkeit auch unter geänderten Rahmenbedingungen anwenden können.

Um die Stufe des tiefsten Verständnis zu erreichen, ist die Interaktion mit einer Sicherheitslücke selbst angebracht, etwa durch das Verteidigen oder Angreifen eines Systems mit einer Sicherheitslücke. Hat eine Angreifer:in es geschafft, eine Sicherheitslücke selbst auszunutzen, hat sie bewiesen das Wissen nicht nur analysiert, sondern auch umfassend bewertet sowie mit dem erfolgreichen Angriff etwas erschaffen zu haben. Selbst wenn dies

---

<sup>2</sup>Vgl. „Active Student“, „Students Decide“, „Honor Questions“, „Test Tube“, „Try It Yourself“, „Prefer Writing“, „Role Play“, „War Game“, „Invisible Teacher“, in [7] J. Eckstein, J. Bergin, und H. Sharp, „Patterns for Active Learning“, in *Proceedings of PLOP*, 2002..

nur in einer simulierten Umgebung geschehen ist, so darf trotzdem davon ausgegangen werden, dass das zugrundeliegende Problem wirklich verstanden wurde.

## 2.4 Serious Games ermöglichen Interaktion

Die skizzierte Form der Interaktion kann der Kategorie *serious game* zugeordnet werden. Allgemein werden diese entwickelt, um über reale Themen wie Gesundheitsfürsorge, Sicherheit oder Umweltfragen aufzuklären, zu schulen oder zu informieren [8]. Wenn serious games das Lernen komplexer und technischer Themen erleichtern, können Konzepte der IT-Sicherheit ebenfalls mit Techniken aus diesem Bereich vermittelt werden. In Lernumgebungen für konkrete zu lernende Konzepte der IT-Sicherheit stellen Spieler sich auf diese Weise realistischen Herausforderungen. [5]

Werden auf diese Art Techniken der IT-Security vermittelt, findet dies oft im Rahmen von *capture the flag* statt, einer Art IT-Security-Wettbewerb. Hier gibt es verschiedene Typen von Aufgaben (*challenges*), beispielsweise im Stil der US-Amerikanischen Quizshow *Jeopardy*, bei der es für gelöste Aufgaben in unterschiedlichen Kategorien Punkte zu gewinnen gibt [4]. Aufgaben in der Kategorie „Web“ beispielsweise bestehen üblicherweise aus einem Webserver, der den Spielern eine Website mit einer oder mehreren absichtlich platzierten Schwachstellen zur Verfügung stellt. Diesen Webserver gilt es anzugreifen, indem die Sicherheitslücke(n) identifiziert und ausgenutzt werden.

Es ist zu beobachten, dass Teilnehmende großes Interesse am wettbewerbsorientierten Lernen zeigen [9] und capture-the-flag-Aufgaben die Interaktion von Lernenden mit dem Lerngegenstand erhöhen, was wiederum zu besser entwickelten Fähigkeiten im Umgang mit Sicherheitslücken führt [10].

Capture the Flag als Eigenart der Serious Games bieten Teilnehmenden eine Möglichkeit in einer extra hierfür bereitgestellten Umgebungen den Umgang mit Sicherheitslücken zu erlernen. Sie sind also ein geeignetes Mittel, um Kenntnisse im Bereich der IT-Sicherheit zu vermitteln und nachhaltig zu festigen.

## 3 Interaktive Lernumgebung

Im Folgenden wird eine Lernumgebung aufgebaut, die die Interaktion mit einem weit verbreitetem und für Programmierer:innen direkt relevanten IT-Sicherheitsproblem

ermöglicht. In diesem serious game besteht die Aufgabe darin eine bewusst platzierte Schwachstelle auszunutzen, was ein tiefes Verständnis für die Problematik fördern soll.

## 3.1 Lernziele

Teilnehmer:innen sollen diese Sicherheitslücken zukünftig kennen. Sie sollen verstanden haben, dass es Konsequenzen haben wird, wenn eine Angreifer:in diese Lücke ausnutzt. Eigener Code sollte zukünftig frei von diesem Problem sein. Sie sollen in der Lage sein eigenen oder fremden Code daraufhin zu analysieren ob er diese oder eine ähnliche Sicherheitslücke enthält. Sie sollten bewerten, ob verwundbarer Code ausgenutzt werden kann. In einer etwaigen Diskussion sollen sie ihr Gegenüber davon überzeugen können, dass diese Lücke entfernt werden muss.

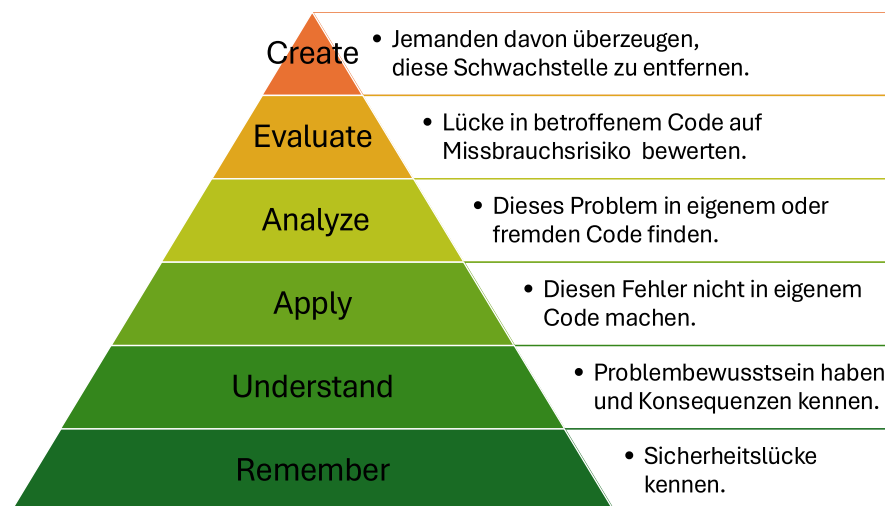


Abbildung 2: Lernziele in Relation zu Blooms Pyramide, nach [3]. Eigene Darstellung

### 3.1.1 Broken Access Control

Schwachstellen werden von der MITRE-Organisation in einer „Common Weakness Enumeration“-Liste (CWE) gesammelt. [11] In den OWASP „Top 10 Web Application Security Risks“ werden Sicherheitslücken kategorisiert und diese Kategorien werden wiederum nach Inzidenzrate geordnet. [6] Auf Platz 1 der schwerwiegendsten Sicherheitsrisiken im Betrachtungszeitraum<sup>3</sup> 2017 – 2021 befinden sich Fehler der Kategorie „Mangelhafte Zugriffskontrolle“. Zugriffskontrollen bezeichnet Mechanismen die sicher-

---

<sup>3</sup>Die nächste Ausgabe der „Top 10 Web Application Security Risks“ wurde für „die erste Jahreshälfte 2025“ angekündigt. [12]



stellen, dass Nutzer:innen nur innerhalb ihrer vorgesehenen Berechtigungen handeln können. Werden diese Zugriffskontrollen mangelhaft implementiert, kann eine erfolgreiche Ausnutzung dazu führen, dass eine Angreifer:in Daten unbefugt lesen, ändern oder zerstören kann oder eine Anwendung dazu bringen kann, Dinge zu tun, die die Nutzer:in nicht können sollte. [13]

## 3.2 Szenario

Die Entstehung der Umgebung soll wenigstens grob der Entstehung einer realen Anwendung folgen. In der anzugreifenden Umgebung soll sich eine fiktive Anwendung befinden, die an und für sich korrekt funktioniert, aber unter Ausnutzung der Sicherheitslücken zu einem (aus Sicht einer fiktiven Organisation) fehlerhaften Verhalten gebracht werden kann.

Für dieses Szenario wird von einer Entwickler:in ausgegangen, die eine Kommunikationsplattform betreiben möchte. Angelehnt an die bestehenden Plattform „Reddit“, bei der sich Nutzer:innen zu vielen verschiedenen Themen öffentlich austauschen können, soll die neue Plattform einen themenbezogenen Austausch in vertraulichen Kleingruppen ermöglichen. Diese Kleingruppen sind im Gegensatz zu Reddit nicht öffentlich, sondern können nur auf Einladung eines Mitglieds betreten werden.

### 3.2.1 Glesn

Die Entwickler:in kreiert die Konkurrenzplattform „Glesn“<sup>4</sup>, welche Anwender:innen verspricht, sich innerhalb eines geschützten „Space“ mit Anderen austauschen zu können. Innerhalb der Spaces können Anwender:innen „Artikel“ schreiben, die von Anderen nur gelesen werden können, wenn diese selbst Teil des Space sind, in dem der Artikel veröffentlicht wurde.

### 3.2.2 Aufbau der Anwendung

Das User-Interface der Website besteht aus einer Kopf-Bereich und einem Bereich für den anzuzeigenden Inhalt. Der Kopfbereich ist aufgeteilt in einen Unter-Bereich für die Account-Verwaltung und einen Unter-Bereich für die Navigation. Anwender:innen können auf der Seite einen Account anlegen, in dem die Berechtigungen (Zugehörigkeit zu Spaces) hinterlegt sind.

---

<sup>4</sup>Bayrisch, vulg. für „gelesen“, Anspielung auf „reddit“, was selbst durch ein Wortspiel mit „I read it on reddit.“ entstanden ist [14].

Eine Einstiegsseite erklärt das Prinzip der Seite, also dass Anwender:innen innerhalb geschützter Spaces Artikel anlegen können und dass andere Anwender:innen diese Artikel nur lesen können, wenn sie zuvor von einer Person innerhalb des Space zu diesem eingeladen wurden.

Über die Navigation kann jeweils eine Übersicht aller für den verwendeten Account sichtbaren Spaces und Artikel aufgerufen werden.

Die Artikel bestehen aus einer Überschrift, der Information, welche Autor:in den Artikel verfasst hat, welchen Spaces der Artikel zugewiesen ist und dem eigentlichen Inhalt des Artikels.

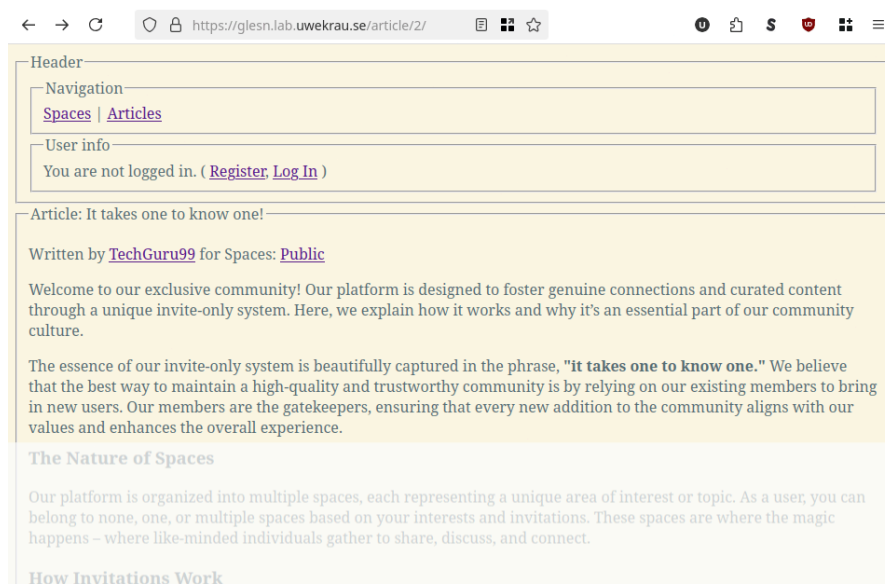


Abbildung 3: Screenshot der Website. In einem öffentlich einsehbareren Artikel wird das Funktionsprinzip erklärt.

Zu einzelnen Autor:innen kann eine Übersicht ihrer Artikel aufgerufen werden. Für angemeldete Anwender:innen besteht außerdem die Möglichkeit, die ausgewählte andere Person zu einem der Spaces einzuladen.



Abbildung 4: Screenshot der Website mit Auflistung der von einer Person verfassten Artikel. Die Person kann zu einem Space eingeladen werden.

## 3.3 Sicherheitslücken in der Anwendung

### 3.3.1 Auslesen von Meta-Daten trotz fehlender Zugriffsberechtigung

Programmierer:innen von Web-Anwendungen unterliegen häufig der falschen Annahme, dass die Bedienung nur mit den von der Anwendung selbst bereitgestellten Mitteln möglich ist. [15] Im Beispiel der Glesn-Anwendung werden sowohl in der Spaces-, als auch in der Artikel-Übersicht nur Links zu Ressourcen-Übersichten aufgelistet, auf die die Anwender:in Zugriff hat. Die vergebenen Berechtigungen (Zugehörigkeit zum Space) wird an dieser Stelle respektiert. Über eine direkte Anfrage ist jedoch ein gezielter Zugriff auf die Ressourcen-Übersichtsseite einer ausgewählten Ressource möglich. Wird in der Glesn-Anwendung eine Artikel-Ansicht direkt aufgerufen, wird auch auf der Übersichtsseite noch einmal die Berechtigung überprüft. Der Inhalt des Artikels bleibt also trotzdem geschützt. Die Übersichts-Seite zeigt jedoch Meta-Daten über den Artikel an: Angezeigt werden die Spaces, denen dieser Eintrag zugewiesen ist und die Autor:in des Artikels. Da die Personen-Übersicht nur Artikel anzeigt, zu denen eine Anwender:in Zugriff hat, sich jedoch über den genannten Umweg trotzdem zumindest alle Artikel-Überschriften zu Autor:innen zuweisen lassen, ist dies als Informationsleck [16] zu betrachten.

### 3.3.2 Rechteausweitung

Wie beschrieben bietet die Anwendung die Möglichkeit, andere Benutzer:innen zu Spaces einzuladen. Durch eine erste Berechtigungsprüfung sind in der grafischen Oberfläche in einer Auswahl-Liste für die Anwender:in nur die Spaces zum Einladen verfügbar, zu

denen Zugriff besteht. Eine Angreifer:in kann die Oberfläche jedoch umgehen und beim Web-Server direkt eine Einladung anfragen. Wieder wird die Annahme ausgenutzt, dass die vom Web-Server angebotenen Endpunkte ausschließlich wie von der Entwickler:in vorgesehen benutzt werden [15]. In der Beispiel-Anwendung wurde davon ausgegangen, dass der Endpunkt durch die erste Berechtigungsprüfung bereits geschützt ist. In diesem Szenario wird deshalb fehlendes Bewusstsein unterstellt, dass vor dem erfolgreichen Aussprechen einer Einladung zu einem Space überprüft werden muss, ob die einladende Person überhaupt selbst über die notwendige Berechtigung verfügt.

Durch die fehlende Autorisationsprüfung werden mehrere Schutzziele der Anwendung gefährdet: Eine Angreifer:in kann eine privilegierte Funktion aufrufen, die Daten in ihrem Sinne verändert. Auf diese Weise erhält sie Zugriff auf geschützte Informationen. Durch Verletzung der Integrität werden Zugriffskontrollen umgangen und die Vertraulichkeit der Anwendung verletzt. [17]

## 3.4 Angedachter Lösungsweg

Die Challenge lässt sich lösen, indem die innerhalb der Anwendung platzierten Schwachstellen ausgenutzt werden. Die Sicherheitslücken müssen miteinander kombiniert werden, um zum Ziel zu kommen. Es sind aber keine besonderen Angriffswerkzeuge vonnöten, ein üblicher Webbrowser reicht aus. Mit dem Werkzeug der „Netzwerkanalyse“, können in modernen Browsern die einzelnen Aufrufe auf technischer Ebene protokolliert, eingesehen und erneut versendet werden. Einige Web-Browser (z.B. Firefox) erlauben von hier auch vor dem Wiederversenden eine Manipulation der Anfrage.

Als erstes wird davon ausgegangen, dass Angreifer:innen die Anwendung auskundschaften. Hierbei sollte schon ohne Account auffallen, dass die öffentlich sichtbaren Spaces und Artikel anhand ihrer ID aufgerufen werden. Der sichtbare Space „Public“ hat die ID 2 und im Browser wird als Navigationspfad `/space/2/` angezeigt. Der darin abgelegte Artikel hat die Artikel-ID 2 und wird über `/article/2/` aufgerufen. Die Oberfläche zeigt unter anderem auch einen Link zur Übersichtsseite der Autor:in des Artikels an, die sich in diesem Beispiel über `user/12/` aufrufen lässt. Abbildung 5 zeigt einen Ausschnitt der Netzwerkanalyse.

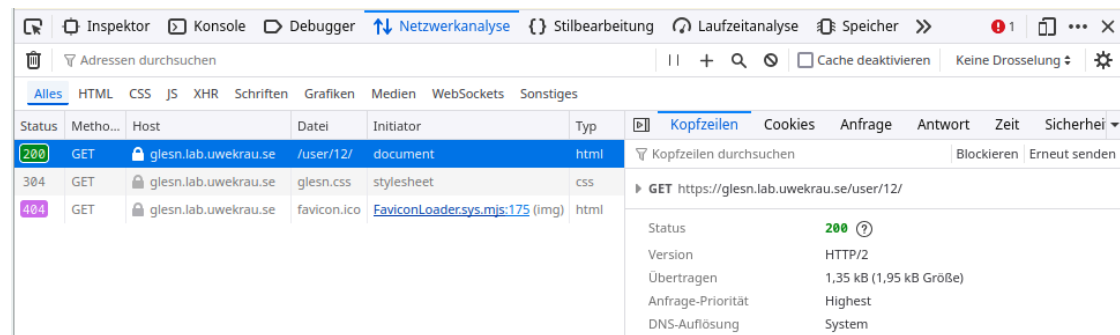


Abbildung 5: Screenshot der Netzwerkanalyse: Erfolgreicher GET Aufruf auf /user/12/.

Dass die ersten sichtbaren Einträge die ID 2 haben, soll als Hinweis verstanden werden, dass es unter den weiteren vorhandenen Spaces, Artikel und User-Accounts auch jeweils Objekte mit der ID 1 gibt, die von besonderem Interesse sein könnten. Durch einen direkten Aufruf der Ressourcen, in Kombination mit der Anzeige der unzureichend geschützten Meta-Daten, können Informationen über weitere Spaces, Artikel und Autor:innen enumeriert werden. So enthalten die Beispieldaten unter anderem ein Account mit dem Namen „admin“, ein Space dessen Name „!!! ADMIN ONLY !!!“ lautet und ein Artikel mit der Überschrift „Note FOR ADMINS ONLY !!!“. Die sehr deutlichen Namen sollen wieder ein Hinweis auf das zu erreichende Ziel sein.

Nun sollte die nächste zur Verfügung stehende Funktionalität der Website verwendet werden: Die Registration eines neuen Accounts. Mit diesem Account sind zwar keine zusätzlichen Spaces oder Artikel sichtbar, aber auf der Informationsseite zu anderen Autoren erscheint nun – wie zuvor in Abbildung 4 gezeigt – die Möglichkeit, die andere Person zu Spaces einzuladen. Mit Hilfe der Netzwerkanalyse kann dem Protokoll entnommen werden, dass eine Einladung zu einem Space auf technischer Ebene durch ein POST-Request angefordert wird. Dieser Anfrage wird als Argument die ID des Spaces über den Parameter `space_id` übergeben, wie in Abbildung 6 dargestellt.

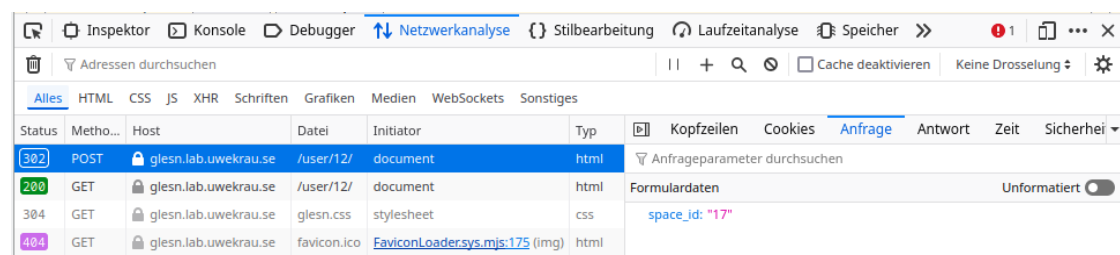


Abbildung 6: Screenshot der Netzwerkanalyse: Erfolgreicher POST Aufruf auf /user/12/ mit Argument `space_id: "17"`.

Der angedachte Lösungsweg sieht vor, dass eine Angreifer:in nun versuche sollte, die Parameter der Anfrage zu manipulieren und zu probieren, verschiedenen Accounts zu verschiedenen Spaces einzuladen. Aufgrund der Sicherheitslücke der fehlenden Berechtigungsprüfung wird dies gelingen und die Angreifer:in kann sich selbst durch eine manipulierte `POST`-Anfrage zu dem Space mit der `space_id`: "1" einladen. Da sie nun Mitglied des Spaces ist, also ihre eigenen Zugriffsrechte erhöht hat, kann sie über die reguläre Benutzeroberfläche auch den Inhalt der eigentlich geschützten Notiz lesen.

## 3.5 Beta-Test

Während der Entwicklung der Challenge wurde ein Beta-Test durchgeführt, bei dem sich freundlicherweise mehrere Mitglieder des lokalen CTF-Teams Cyclopropenylidene (C3H2) bereit erklärt haben, eine Vorabversion der Aufgabe zu lösen. Aus den gesammelten Rückmeldungen sind Zwei Punkte besonders nennenswert:

In einer früheren Version wurden in die Datenbank der Beispieldaten zunächst die Artikel eingefügt, dann erst der zu findende Artikel mit den versteckten Informationen (siehe Abschnitt 3.4). Dies führte dazu, dass dieser Artikel eine unvorhersehbare ID wie 17 oder 19 hatte, wodurch wiederum das Ziel nicht so klar war. Einer der Beta-Tester hat angemerkt, dass die IDs als eine Art Hinweis verwendet werden könnten, wie dann später umgesetzt.

Außerdem wurde ein technisches Problem aufgedeckt: Ein Beta-Tester hat nicht nur einen Account angelegt und (wie angedacht) diesem Account Zugriff auf den versteckten Space gegeben, sondern mit Hilfe eines http replay Proxy die Sicherheitslücke so ausgenutzt, dass alle Accounts Zugriff auf den versteckten Space hatten. Das hat dafür gesorgt, dass eine Beta-Testerin, die sich zuvor einen Account angelegt hat, aber die Sicherheitslücke noch nicht gefunden hat, plötzlich ohne eigenes Zutun Zugriff auf den versteckten Space hatte. Für diese Challenge ist es also notwendig, dass sie pro Person/Team als eigene isolierte Instanz bereitgestellt werden muss.

## 3.6 Durchführung Workshop

Während der sogenannten [Workshop Week] an der Hochschule für Angewandte Wissenschaften (HAW) Hamburg im September 2024 wurden mit Hilfe diese Challenge in einer halbtägigen Veranstaltung ca. 12 Teilnehmer:innen in das Problemfeld der mangelhaften

Zugriffskontrolle eingeführt. Es wurde zwar keine strukturierte Evaluation durchgeführt, aber aus Reaktionen während der Veranstaltung und in unstrukturierten Gesprächen nach der Veranstaltung wurde ersichtlich, dass wenigstens einzelne Teilnehmer:innen gelernt haben, wie leicht http-Anfragen wie `GET` und `POST` manipuliert werden können und wie wichtig daher eine serverseitige Überprüfung der Nutzereingaben ist. Zusätzlich dazu wurde ein deutliches Interesse an dieser Lernform zu IT-Sicherheitsthemen geäußert.

## 4 Fazit & Ausblick

### 4.1 Fazit

In dieser Arbeit wurde aufbauend auf der didaktischen Basis von Blooms Taxonomie eine Methode gesucht, mit dem Ziel das Bewusstsein für Sicherheitsprobleme bei der Programmierung von IT-Systemen zu steigern. Während der ICAP-Ansatzes eine möglichst starke Interaktion mit dem Lerngegenstand nahelegt, interagieren Anwender:innen vertieft mit Unterhaltungsmedien, weshalb eine Methode aus dem Unterhaltungsbereich entliehen wurde. Als Teil eines Serious Games wurde eine interaktive Lernumgebung entwickelt, anhand derer Teilnehmer:innen eines Workshops selbst erleben konnten, welche Auswirkungen Nachlässigkeit bei der Programmierung von IT-Systemen haben kann.

In dem fiktiven Szenario einer Programmierer:in, die die Zugriffskontrolle einer Web-Anwendung unzureichend abgesichert hat, konnten die Teilnehmer:innen in die Rolle einer Angreifer:in schlüpfen und mit frei zugänglichen Werkzeugen diese Sicherheitslücken selbst ausnutzen. Aus den Rückmeldungen der Teilnehmer:innen war Lernerfolg und gesteigertes Interesse für interaktives Lernen von IT-Sicherheitsthemen (ctf) ersichtlich.

### 4.2 Ausblick

Aufbauend auf und ergänzend zu diesem Beispiel sollten weitere Empfehlungen für Verbesserungen und Erkenntnisse zur Gestaltung und Entwicklung effektiverer Serious Games [18] betrachtet werden, um das Sicherheitsbewusstsein von Programmierer:innen zu steigern. Als konkretes Beispiel könnten Evaluations-Methoden zur User Experience aus der allgemeinen Spiele-Entwicklung oder dem Übergeordneten Feld der „serious games“ auch auf die entwickelte Challenge adaptiert werden [19].

Die entwickelte Challenge sollte geprüft werden, ob sie formulierten Anforderungen an Lernumgebungen für Serious Games [20] entspricht. Bei einer weiteren zu entwickelnden Challenge sollten die dort formulierten Anforderungen von Anfang an integriert werden. Stehen mehrere Challenges zur Verfügung, können diese für ein „Capture The Flag“-Event verwendet werden. Ein solches Event kann anhand eines festgelegten Rahmenwerks mit anderen Events verglichen werden [21].



# Bibliographie

- [1] „Cost of a Data Breach 2024 | IBM“. Zugegriffen: 14. April 2025. [Online]. Verfügbar unter: <https://www.ibm.com/reports/data-breach>
- [2] M. T. H. Chi und R. Wylie, „The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes“, *Educational Psychologist*, Bd. 49, Nr. 4, S. 219–243, Okt. 2014, doi: 10.1080/00461520.2014.965823.
- [3] L. W. Anderson und D. R. Krathwohl, Hrsg., *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*, Complete ed. New York: Longman, 2001.
- [4] CTFtime team, „What Is Capture The Flag?“. Zugegriffen: 24. Februar 2025. [Online]. Verfügbar unter: <https://ctftime.org/ctf-wtf/>
- [5] G. Costa und M. Ribaud, „Designing a Serious Game for Cybersecurity Education“, *Software Engineering for Games in Serious Contexts*. Springer Nature Switzerland, Cham, S. 265–290, 2023. doi: 10.1007/978-3-031-33338-5\_12.
- [6] „OWASP Top 10:2021“. Zugegriffen: 25. Februar 2025. [Online]. Verfügbar unter: <https://owasp.org/Top10/de/>
- [7] J. Eckstein, J. Bergin, und H. Sharp, „Patterns for Active Learning“, in *Proceedings of PLOP*, 2002.
- [8] K. M. L. Cooper und A. Bucchiarone, Hrsg., *Software Engineering for Games in Serious Contexts*. Cham: Springer Nature Switzerland, 2023. doi: 10.1007/978-3-031-33338-5\_12.
- [9] A. Siami Namin, Z. Aguirre-Muñoz, Texas Tech University, und K. Jones, „Teaching Cyber Security through Competition An Experience Report about a Participatory Training Workshop“, in *7th Annual International Conference on Computer Science Education: Innovation & Technology (CSEIT 2016)*, Global Science & Technology Forum (GSTF), Okt. 2016. doi: 10.5176/2251-2195\_CSEIT16.39.
- [10] K. Leune und S. J. Petrilli, „Using Capture-the-Flag to Enhance the Effectiveness of Cybersecurity Education“, in *Proceedings of the 18th Annual Conference on Information Technology Education*, Rochester New York USA: ACM, Sep. 2017, S. 47–52. doi: 10.1145/3125659.3125686.

- [11] MITRE Corporation, „Common Weakness Enumeration - CWE Version 4.16“. Zugegriffen: 25. Februar 2025. [Online]. Verfügbar unter: [https://cwe.mitre.org/data/published/cwe\\_v4.16.pdf](https://cwe.mitre.org/data/published/cwe_v4.16.pdf)
- [12] „OWASP Top Ten | OWASP Foundation“. Zugegriffen: 25. Februar 2025. [Online]. Verfügbar unter: <https://owasp.org/www-project-top-ten/>
- [13] „A01:2021 – Mangelhafte Zugriffskontrolle - OWASP Top 10:2021“. Zugegriffen: 26. Februar 2025. [Online]. Verfügbar unter: [https://owasp.org/Top10/de/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/de/A01_2021-Broken_Access_Control/)
- [14] „Faq - Reddit.Com“. Zugegriffen: 26. Februar 2025. [Online]. Verfügbar unter: [https://web.archive.org/web/20201127015655/http://www.reddit.com/wiki/faq#wiki\\_what\\_does\\_the\\_name\\_.22reddit.22\\_mean.3F](https://web.archive.org/web/20201127015655/http://www.reddit.com/wiki/faq#wiki_what_does_the_name_.22reddit.22_mean.3F)
- [15] „CWE - CWE-425: Direct Request ('Forced Browsing') (4.16)“. Zugegriffen: 25. Februar 2025. [Online]. Verfügbar unter: <https://cwe.mitre.org/data/definitions/425.html>
- [16] „CWE - CWE-200: Exposure of Sensitive Information to an Unauthorized Actor (4.16)“. Zugegriffen: 25. Februar 2025. [Online]. Verfügbar unter: <https://cwe.mitre.org/data/definitions/200.html>
- [17] „CWE - CWE-862: Missing Authorization (4.16)“. Zugegriffen: 25. Februar 2025. [Online]. Verfügbar unter: <https://cwe.mitre.org/data/definitions/862.html>
- [18] V. Wanick, J. Stallwood, und G. Xavier, „Future Directions in Games for Serious Contexts: A Conversation About Transferability“, *Software Engineering for Games in Serious Contexts*. Springer Nature Switzerland, Cham, S. 137–153, 2023. doi: 10.1007/978-3-031-33338-5\_7.
- [19] S. Kirginas, „User Experience Evaluation Methods for Games in Serious Contexts“, *Software Engineering for Games in Serious Contexts*. Springer Nature Switzerland, Cham, S. 19–42, 2023. doi: 10.1007/978-3-031-33338-5\_2.
- [20] T. Espinha Gasiba, K. Beckers, S. Suppan, und F. Rezabek, „On the Requirements for Serious Games Geared Towards Software Developers in the Industry“, in *2019 IEEE 27th International Requirements Engineering Conference (RE)*, Jeju Island, Korea (South): IEEE, Sep. 2019, S. 286–296. doi: 10.1109/RE.2019.00038.
- [21] R. Raman, S. Sunny, V. Pavithran, und K. Achuthan, „Framework for Evaluating Capture the Flag (CTF) Security Competitions“, in *International Conference for*

*Convergence for Technology-2014*, Pune, India: IEEE, Apr. 2014, S. 1–5. doi: 10.1109/I2CT.2014.7092098.